

SHIFTAHOLIC

Proyecto realizado por:

Jordi Rivas Ruiz
Daniel Cruz Morales
Marco A. Cocera López

SHIFTAHOLIC

Análisis

SHIFTAHOLIC

Qué es Shiftaholic

- Aplicación web que permite gestionar los turnos
- Aplicación móvil que permite solicitar y recibir peticiones de cambio de turnos

Para quién es Shiftaholic

- Empresas con gestión de turnos compleja en la que los cambios de turno son habituales. Colectivos como médicos, enfermeros, controladores...

SHIFTAHOLIC

Tecnologías

Android (Android Studio)

- Realm
- EventBus
- Retrofit

JavaEE (Eclipse)

- Jersey

Web (Eclipse)

- Bootstrap
- JQuery

SHIFTAHOLIC

Casos de uso

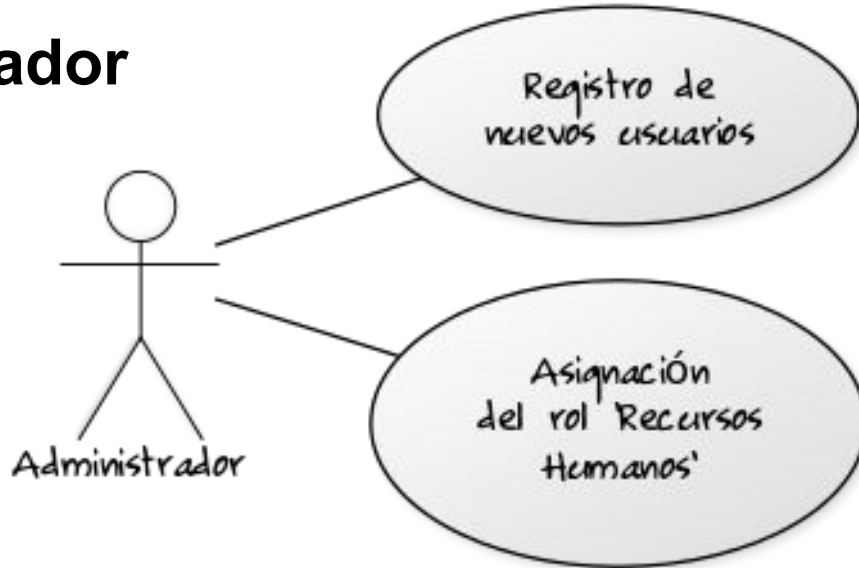
SHIFTAHOLIC

Roles

- Administrador
- Recursos Humanos
- Usuario

SHIFTAHOLIC


Administrador



SHIFTAHOLIC

Administrador

Shiftaholic






















[admin](#)

Usuarios

Ver

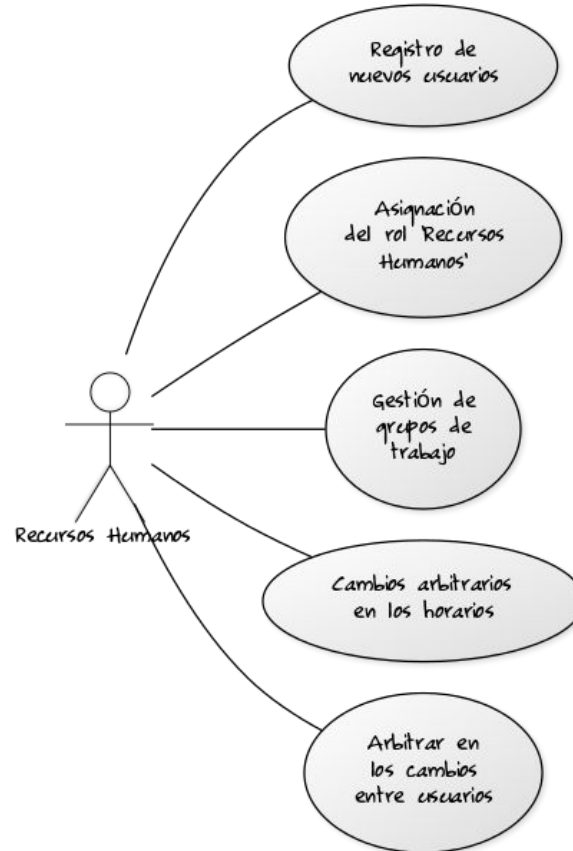
Registro

Archivos

ID	DNI	Nombre	Apellido	2º Apellido	Email	Activo					
malfs	38561924I	Mónica	Alfárez	Siruella	muchomo@hotmail.com	✓					
málv	32562563Y	Miguel	Álvarez		miguelal@gmail.com	✓					
rarm	2566315p	Ricard	Arroyo	Martín	rarma@gmail.com	✓					
jcocs	38561923P	Jessica	Cocera	Soria	mantisbboy@gmail.com	✓					
mcocl	48007284M	Marco A.	Cocera	López	marcoacolo@gmail.com	✓					

SHIFTAHOLIC

Recursos Humanos



SHIFTAHOLIC

Recursos Humanos

Shiftaholic

Subir foto

Salir

arivr

Usuarios

Grupos

Ver grupos

Gestionar turnos

Ver peticiones

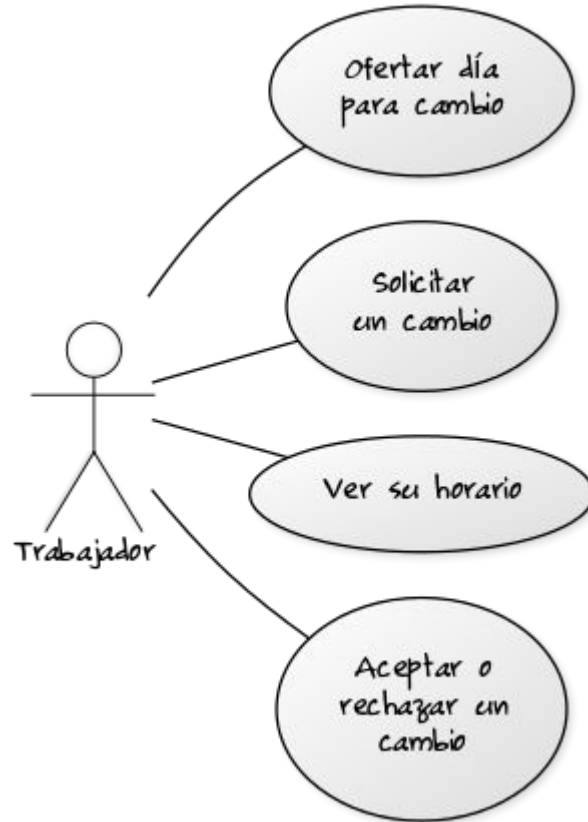
Cambio manual

Configurar turnos

ID	DNI	Nombre	Apellido	2º Apellido	Email	Gr
malfs	38561924I	Mónica	Alfárez	Siruela	muchomo@hotmail.com	Mañana ▼
málv	32562563Y	Miguel	Álvarez		miguelal@gmail.com	Indefinido ▼
rarm	2566315p	Ricard	Arroyo	Martín	rarrma@gmail.com	Indefinido ▼
jcocs	38561923P	Jessica	Cocera	Soria	mantisbboy@gmail.com	Indefinido ▼
mcocl	48007284M	Marco A.	Cocera	López	marcoacolo@gmail.com	Indefinido ▼
ccrul	38561924F	Cristian	Cruz	Lisonja	crispeli@gmail.com	Indefinido ▼
dcrum	38561924Z	Daniel	Cruz	Morales	dani.cruz.morales@gmail.com	Indefinido ▼

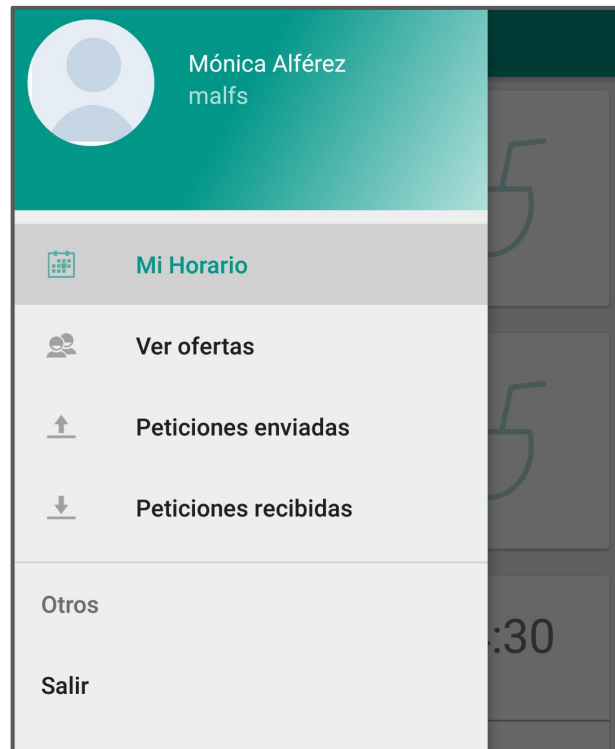
SHIFTAHOLIC

Usuario (*trabajador*)



SHIFTAHOLIC

Usuario (*trabajador*)



SHIFTAHOLIC

Funcionamiento

SHIFTAHOLIC

Descripción de la Base de datos

users

shift

shift_type

request_shift

SHIFTAHOLIC

Base de datos - users

Columna	Tipo	Modificadores
user_id	character varying	not null
external_id	character varying	not null
name	character varying	not null
surname1	character varying	not null
surname2	character varying	
email	character varying	not null
creation_time	bigint	not null
removal_time	bigint	

SHIFTAHOLIC

Base de datos - shift_type

Columna	Tipo	Modificadores
id	integer	not null
name	character varying	not null
real_hours	numeric(4,2)	not null
computed_hours	numeric(4,2)	not null
ini_time	character varying(5)	not null
end_time	character varying(5)	not null
shift_generator	integer[]	not null
week_day	integer	not null
next_assignement	bigint	

SHIFTAHOLIC

Base de datos - shift

Columna	Tipo	Modificadores
id	integer	not null
user_id	character varying	not null
shift_type	integer	not null
ini_time	character varying(5)	not null
end_time	character varying(5)	not null
computed_hours	numeric(4,2)	not null
day	bigint	not null
day_type	character varying	not null
worked	boolean	not null
changeable	boolean	
pending	boolean	
request_shift_id	integer	
last_modification	bigint	

SHIFTAHOLIC

Base de datos - request_shift

Columna	Tipo	Modificadores
id	integer	not null
own_shift	integer	not null
other_shift	integer	not null
own_free_shift	integer	
other_free_shift	integer	
requester_user	character varying	not null
requested_user	character varying	not null
rrhh_user	character varying	
state	integer	not null
reason	text	
last_modification	bigint	

SHIFTAHOLIC

Descripción del Servidor

shiftaholic/web/app

shiftaholic/rest

shiftaholic/password/restoration

shiftaholic/web/mediaserver

SHIFTAHOLIC

Funcionamiento del servidor *web*

SHIFTAHOLIC

Funcionamiento del servidor - web




SHIFTAHOLIC

Funcionamiento del servidor - web

Solicitud del cliente

```
function fetchData(contentid, targetid) {  
    var set;  
    if (targetid) {  
        set = function(data) {  
            $('#'+ targetid).html(data);  
        };  
    }  
    var req = new ContentRequest(contentid);  
    req.access=userAccess;  
    var par = JSON.stringify(req);  
    console.log(req);  
    console.log(par);  
    $.ajax({  
        url : CONTENTURL,  
        type : 'POST',  
        data : par,  
        contentType : "application/json; charset=utf-8",  
        dataType : "html",  
        success : set  
    });  
}
```



```
function ContentRequest(contentid) {  
    this.content = contentid;  
    this.token = sessionStorage.getItem("token");  
}
```

SHIFTAHOLIC

Funcionamiento del servidor - web

Procesado del servidor

```
@WebServlet(name = "app", urlPatterns = { "/web/app" })
public class PagesServer extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private static final String URL_BASE = "/WEB-INF/jsp/";

    private void process(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        // Cargamos la request que viene como json
        ContentRequest contentRequest = new Gson().fromJson(new InputStreamReader(request.getInputStream()),
            ContentRequest.class);

        if (contentRequest == null || !isValidRequest(contentRequest, request)) {
            // TODO que hacer cuando no es valida.
            return;
        }

        dispatchFragment(contentRequest, request, response);
    }
}
```

SHIFTAHOLIC

Funcionamiento del servidor - web

Procesado del servidor

```
private void dispatchFragment(ContentRequest contentRequest, HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {

    String content = contentRequest.getContent();

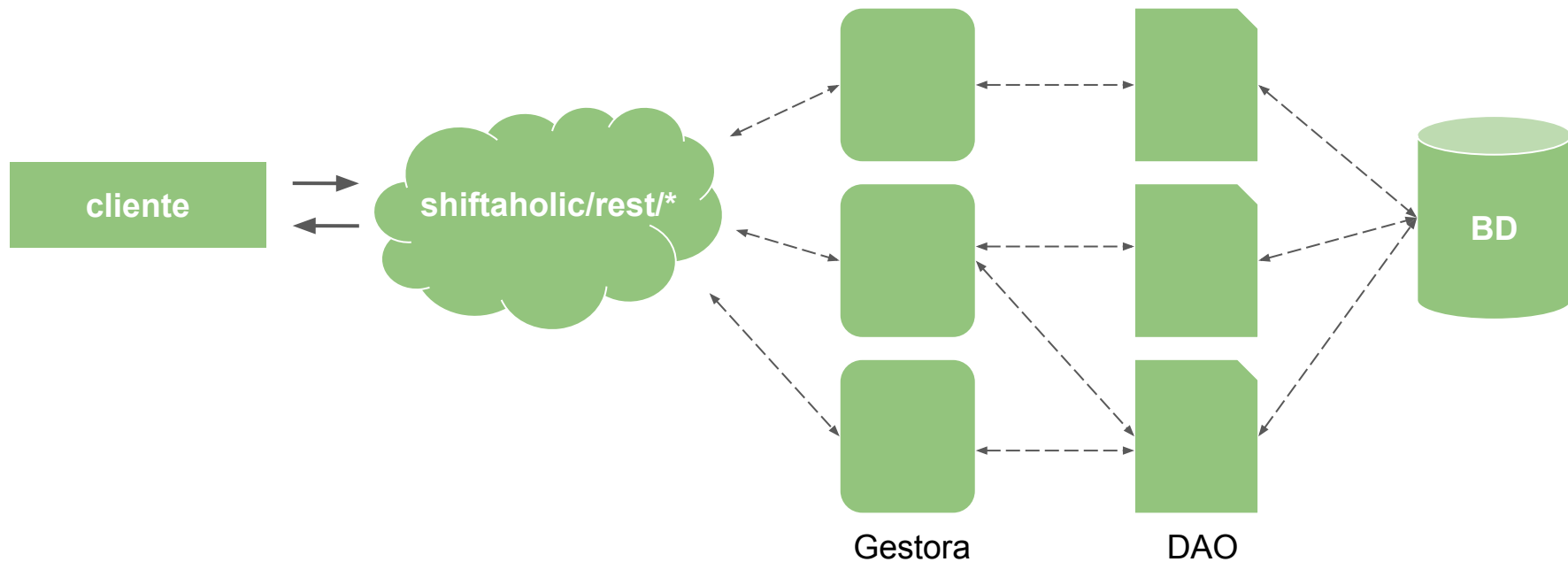
    if (content.equals("app") && contentRequest.getAccess().equalsIgnoreCase("admin")) {
        request.setAttribute("access", "admin");
    }
    request.getRequestDispatcher(URL_BASE + "fragments/" + content + ".jsp")
        .forward(request, response);
}
```


SHIFTAHOLIC

Funcionamiento del servidor ***rest***

SHIFTAHOLIC

Funcionamiento del servidor - rest



SHIFTAHOLIC

Funcionamiento del servidor - rest

```
function doRestPost(req, url, callback, errorCallback) {  
  
    var par = JSON.stringify(req);  
    console.log(req);  
    $.ajax({  
        url : RESTURL + url,  
        type : 'POST',  
        data : par,  
        async : true,  
        contentType : "application/json; charset=utf-8",  
        dataType : "json",  
        success : callback,  
        error : errorCallback  
    });  
}
```

Solicitud del
cliente
web

SHIFTAHOLIC

Funcionamiento del servidor - rest

```
public void doLogin(String username, String password) {
    RestRequest request = new RestRequest();
    request.getParams().put("userid", username);
    request.getParams().put("password", password);

    userServices.postLogin(request).enqueue(new Callback<LoginResponse>() {
        @Override
        public void onResponse(Call<LoginResponse> call, Response<LoginResponse> response) {
            Log.d("doLogin", "response ok - " + response.body());
            EventBus.getDefault().post(response.body());
        }

        @Override
        public void onFailure(Call<LoginResponse> call, Throwable t) {
            Log.e("doLogin", "response ko - " + call.request().url(), t);
            postError(serverError);
        }
    });
}
```

Solicitud del
cliente
android

SHIFTAHOLIC

Funcionamiento del servidor - rest

```
@Path("login")
@POST
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public LoginResponse login(RestRequest request) {
    LoginResponse response = new LoginResponse();
    Map<String, String> params = request.getParams();
    Login login = manager.doLogin(params.get("userid"), params.get("password"));
    String errorCode = login.getErrorCode();
    response.setErrorCode(errorCode);
    if (errorCode.equals(ErrorCodes.ERROR_CODE_NO_ERROR)) {
        response.setLastTime(login.getLastLogin());
        response.setToken(login.getToken());
        response.setUserId(login.getUserId());
        response.setName(login.getName());
        response.setSurname(login.getSurname());
    }
    return response;
}
```

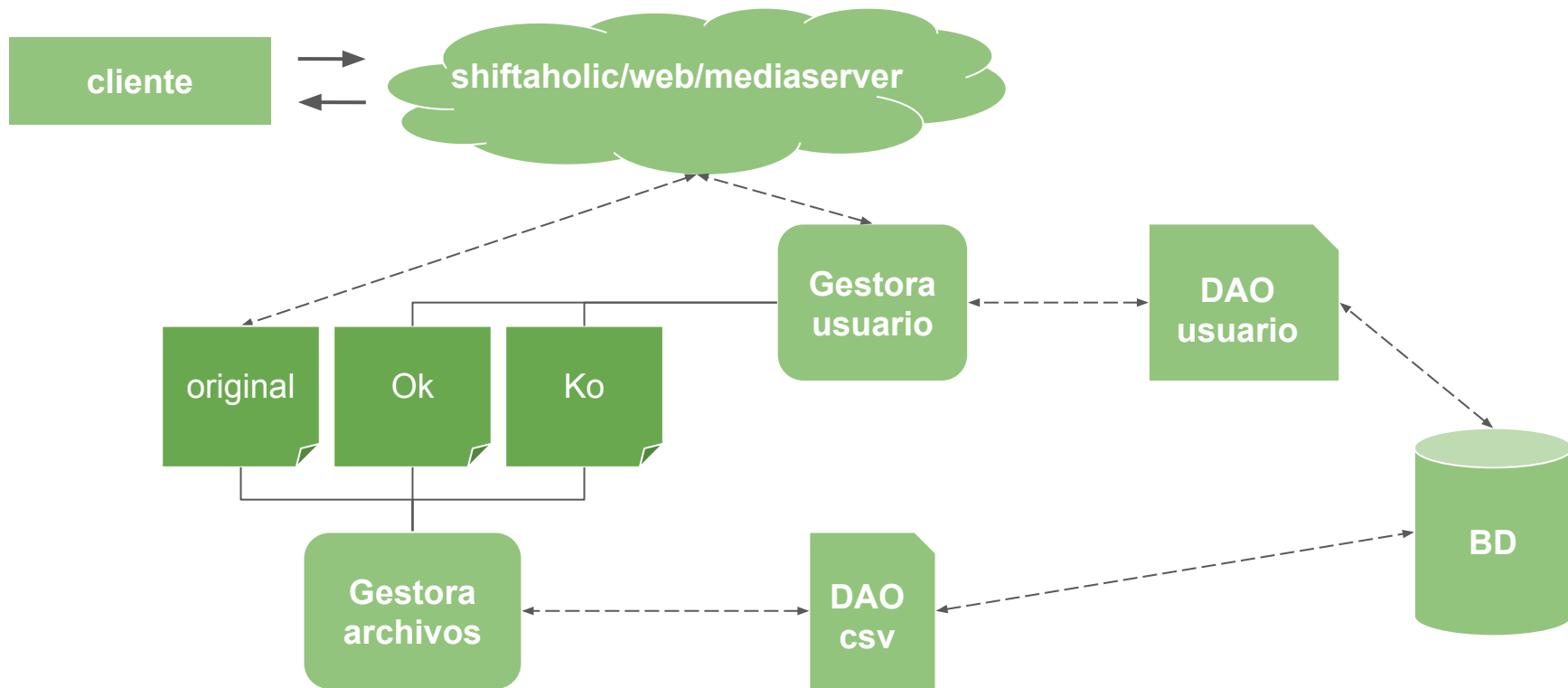
Procesado
del
Servidor

SHIFTAHOLIC

Funcionamiento del servidor
archivos csv

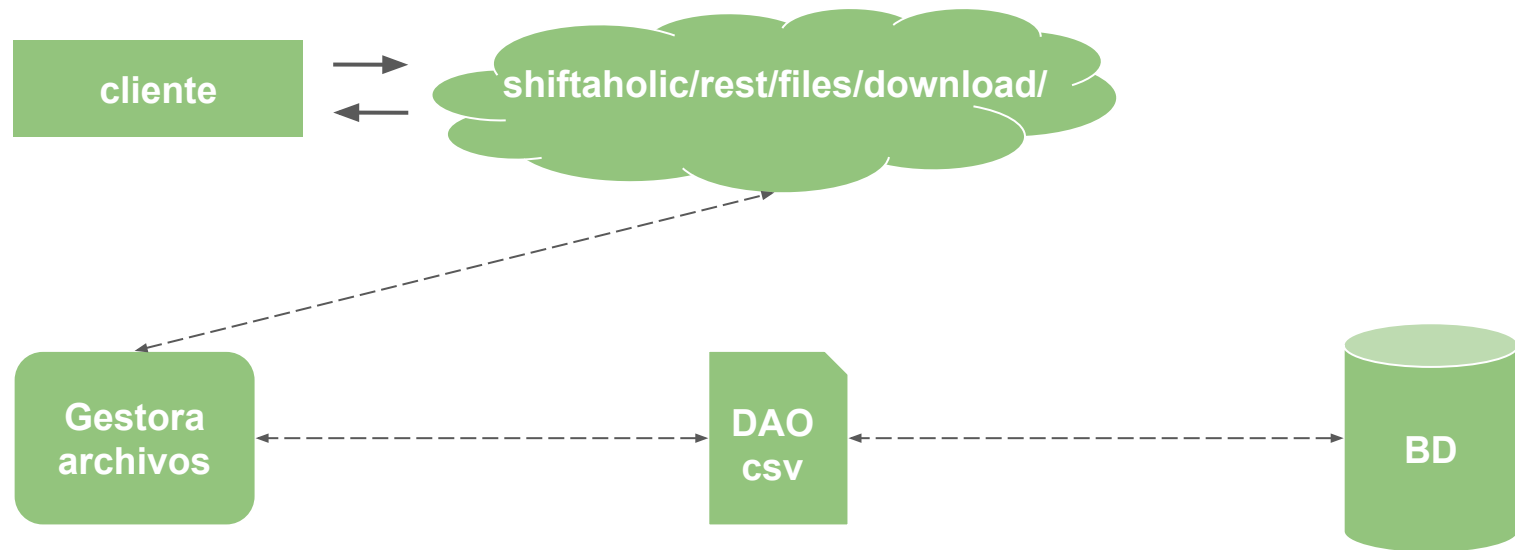
SHIFTAHOLIC

Funcionamiento del servidor - archivos csv (*subida*)



SHIFTAHOLIC

Funcionamiento del servidor - archivos csv (*bajada*)



SHIFTAHOLIC

Funcionamiento del servidor - archivos csv (*bajada*)

```
@GET
@Path("download/{type}/{bundle}/{filename}")
@Produces(MediaType.APPLICATION_OCTET_STREAM)
public Response downloadCSVFile(@QueryParam("token") String token,
    @PathParam("type") String type,
    @PathParam("bundle") String bundle,
    @PathParam("filename") String name) {
    if (!userManager.validToken(token)) {
        return Response.status(Response.Status.UNAUTHORIZED).build();
    }
    FileType ftype = null;
    switch (type) {
        case "registration":
            ftype = FileType.REGISTRATION_FILE;
            break;

        default:
            return Response.status(Response.Status.BAD_REQUEST).build();
    }
    DatabaseBinary file = fileManager.getFile(bundle, name, ftype);
    if (file == null) {
        return Response.status(Response.Status.NOT_FOUND).build();
    } else {
        return Response.ok(file.getStream()).build();
    }
}
```

SHIFTAHOLIC

Instalación

SHIFTAHOLIC

Requisitos del Servidor

- Tomcat v8
- PostgreSQL v9.5

SHIFTAHOLIC

Demo

<http://80.28.157.253:8181/shiftaholic/web/app>